CS4320 Prelim Exam

March 17th, 2017
(50 minutes working time)

Name: _____ Cornell NETID: _____

I understand and will adhere to the Cornell Code of Academic Integrity.

Signature

Maximum number of points possible: 40. This exam counts for 20 % of your overall grade. Questions vary in difficulty. Do not get stuck on one question.

In all problems, whenever you think a problem is underspecified, make assumptions and clearly state them.

Good luck!

Note – you have 50 minutes working time for this exam, NOT 2 hours as on some other prelims.

Part A) SQL Queries and Relational Algebra. (15 points)

Consider the database schema created by the following SQL commands:

```
CREATE TABLE Students(studentID INTEGER PRIMARY KEY,
name VARCHAR(30));

CREATE TABLE Courses(courseID INTEGER PRIMARY KEY,
courseName VARCHAR(30));

CREATE TABLE Taken(courseID INTEGER, studentID INTEGER,
PRIMARY KEY(courseID, studentID),
FOREIGN KEY (courseID) REFERENCES Courses(courseID),
FOREIGN KEY (studentID) REFERENCES Students(studentID));
```

This database contains information about students (table `Students`) and courses (table `Courses`) and assigns each student to the courses this student takes (table `Taken`).

A.1) Write an SQL query retrieving for each course the number of students taking it. (5 points)

```
SELECT C.coursename, C.courseID, COUNT(*)
FROM Courses C, Students S, Taken T
WHERE C.courseID = T.courseID AND T.studentID = S.studentID
GROUP BY C.courseID, C.coursename;
```

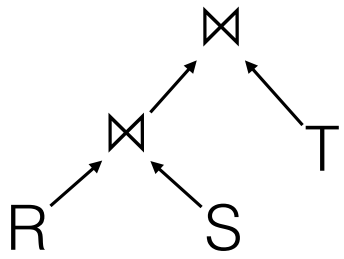
A.2) Write an SQL query retrieving the names of all students who took at least two courses.
(5 points)

```
SELECT S.name, S.studentID
FROM Students S, Taken T, Courses C
WHERE S.studentID = T.studentID AND T.courseID = C.courseID
GROUP BY S.studentID, S.name
HAVING COUNT(*) >= 2;
```

A.3) Write a relational algebra query that retrieves pairs of course IDs such that there is at least one student that takes both courses. Your relational algebra query may only use the projection operator, the selection operator, the cross product, and the renaming operator.
(5 points)

$$\Pi_{T1.courseID,T2.courseID}(\sigma_{T1.studentID=T2.studentID \wedge T1.courseID \neq T2.courseID}(\rho_{(T1,Taken)} \times \rho_{(T2,Taken)}))$$

Part B) Execution Cost Estimation. (15 points)



Consider the query plan above joining three tables (R, S, and T). We assume that a block nested loop join is used for the first join (i.e., the join between R and S) and that an index nested loop join is used for the second join (i.e., the join with T). The index nested loop join uses an unclustered hash index on table T.

We assume that table R contains 50,000 rows, table S contains 10,000 rows, and table T contains 200,000 rows. The selectivity of the first join condition (i.e., the ratio of tuples satisfying the join condition) is 0.001. The second join condition is an equality condition (you do not need its selectivity).

We make the simplifying assumption that 100 rows (no matter from which table or from which join result) fit on one memory page and 100 pages of main memory are available. Assume that the cost of accessing the index and retrieving the corresponding data is always 2 page I/Os. The block nested loop joins uses blocks of size 10 pages for reading the outer relation and intermediate results remain in main memory (no write back to disc) whenever possible.

Calculate the total execution cost of the query plan as the number of page I/Os, without taking into account any cost for writing down the result of the plan to disc. Break down cost into components that are associated with different operations in the query plan and explain each step of your calculation. You can use the following blank page for your calculations.

Note that the query plan (in particular in terms of the decision to use an index nested loop join for the second join) is not necessarily optimal.

Page sizes of input relations:
pages(R) = 50,000 / 100 = 500
pages(S) = 10,000 / 100 = 100
pages(T) = 200,000 / 100 = 2000

Cost of first join (BNL):
500 I/Os (cost of reading outer) +
500/10 (number of iterations) * 100 I/Os (cost of reading inner) =
5500 I/Os

Size of result of first join: 50,000 * 10,000 * 0.001 = 500,000

Cost of second join (IdNL):
500,000 (number of tuples in outer relation) * 2 I/Os (index and data access) =
1,000,000 disc I/Os

Note that the result of the first join does not need to be written back to disc (the result does not fit into main memory but we can use pipelining).

Total cost is 1,005,500 disc I/Os.

Part C) Query Optimization and Duplicate Elimination. (10 points)

C.1) Explain the term “Left-Deep Query Plan”. (5 points)

A query plan where the outer input relation of each join is the result of all previous joins except for the first join.

C.2) Name two different methods for eliminating duplicates and give a one or two sentence description of how they work. (5 points)

Sorting based duplicate elimination: before eliminating duplicates, we sort tuples to place duplicates close to each other. Duplicates form sequences among the sorted tuples and we can detect and eliminate them during a single scan.

Hash based duplicate elimination: we partition tuples according to their hash values to place duplicates in the same bucket. Then we use a second hash function to compare tuples within the same bucket and eliminate duplicates.

CS4320 Prelim Exam

This page will be used for grading your exam. Do not write anything on this page.

SECTION	QUESTION	SCORE	SECTION TOTAL
Part A	**A.1 (Max: 5 points)**		**(Max: 15 points)**
	A.2 (Max: 5 points)		
	A.3 (Max: 5 points)		
Part B	**B.1 (Max: 15 points)**		**(Max: 15 points)**
Part C	**C.1 (Max: 5 points)**		**(Max: 10 points)**
	C.2 (Max: 5 points)		
Total (Max: 40 points)			